

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

THIS PAGE BLANK (USPTO)

09/913708-7CT/EP 00/00314
BUNDESREPUBLIK DEUTSCHLAND

REC'D 12 APR 2000

WIPO PCT

EPO-München
52

29. März 2000

EP 00/314



Bescheinigung

Die Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung eV in München/Deutschland hat eine Patentanmeldung unter der Bezeichnung

"Vorrichtung und Verfahren zum Erzeugen eines Datenstroms und Vorrichtung und Verfahren zum Lesen eines Datenstroms"

am 23. Februar 1999 beim Deutschen Patent- und Markenamt eingereicht.

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

Die Anmeldung hat im Deutschen Patent- und Markenamt vorläufig das Symbol H 03 M 7/40 der Internationalen Patentklassifikation erhalten.

München, den 14. Februar 2000

Deutsches Patent- und Markenamt

Der Präsident

Im Auftrag

zeichnen: 199 07.728.2

Hoß

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

Schoppe & Zimmermann · Postfach 710867 · 81458 München

**Fraunhofer-Gesellschaft
zur Förderung der
angewandten Forschung e. V.
Leonrodstraße 54
80636 München**

Fritz Schoppe, Dipl.-Ing.
Tankred Zimmermann, Dipl.-Ing.

Telefon/Telephone 089/790445-0
Telefax/Facsimile 089/790 22 15
Telefax/Facsimile 089/74996977
e-mail 101345.3117@CompuServe.com

**Vorrichtung und Verfahren zum Erzeugen eines Datenstroms und
Vorrichtung und Verfahren zum Lesen eines Datenstroms**

Postanschrift/Mail address: Postfach/P. O. Box 710867, 81458 München

Kanzleianschrift/Office address: Irmgardstraße 22, 81479 München

Bankverbindung/Bankers: HypoVereinsbank Grünwald, Kontonummer 2960 155 028 (BLZ 700 200 01)

Postgiroamt München, Kontonummer 315 720-803 (BLZ 700 100 80)

USt-Id Nr./VAT Registration Number DE 130575439

Vorrichtung und Verfahren zum Erzeugen eines Datenstroms und Vorrichtung und Verfahren zum Lesen eines Datenstroms

Beschreibung

Die vorliegende Erfindung bezieht sich auf das Codieren mit Codeworten variabler Länge und insbesondere auf gegen Übertragungsfehler robuste Datenströme mit Codewörtern variabler Länge.

Moderne Audiocodierverfahren bzw. -decodierverfahren, die beispielsweise nach dem Standard MPEG-Layer 3 arbeiten, sind in der Lage, die Datenrate von Audiosignalen beispielsweise um einen Faktor 12 zu komprimieren, ohne die Qualität derselben merkbar zu verschlechtern. Um eine derartig hohe Datenratenreduktion zu erreichen, wird ein Audiosignal abgetastet, wodurch eine Folge von zeitdiskreten Abtastwerten erhalten wird. Wie es in der Technik bekannt ist, wird diese Folge von zeitdiskreten Abtastwerten mittels geeigneter Fensterfunktionen gefenstert, um gefensterte Blöcke von zeitlichen Abtastwerten zu erhalten. Ein Block zeitlich gefensterter Abtastwerte wird dann mittels einer Filterbank, einer modifizierten diskreten Cosinustransformation (MDCT) oder einer anderen geeigneten Einrichtung in den Frequenzbereich transformiert, um Spektralwerte zu erhalten, die insgesamt das Audiosignal, d. h. den zeitlichen Ausschnitt, der durch den Block von zeitdiskreten Abtastwerten gegeben ist, im Frequenzbereich darstellen. Üblicherweise werden sich zu 50% überlappende zeitliche Blöcke erzeugt und mittels einer MDCT in den Frequenzbereich transformiert, wodurch aufgrund der speziellen Eigenschaften der MDCT immer beispielsweise 1024 zeitdiskrete Abtastwerte zu 1024 Spektralwerten führen.

Es ist bekannt, daß die Aufnahmefähigkeit des menschlichen Ohrs vom Augenblicksspektrum des Audiosignals selbst ab-

hängt. Diese Abhängigkeit ist in dem sog. psychoakustischen Modell erfaßt, mittels dem es seit längerem möglich ist, abhängig vom augenblicklichen Spektrum Maskierungsschwellen zu berechnen. Maskierung bedeutet, daß ein bestimmter Ton bzw. Spektralanteil verdeckt wird, wenn beispielsweise ein benachbarter Spektralbereich eine relativ hohe Energie besitzt. Diese Tatsache der Maskierung wird ausgenutzt, um die nach der Transformation vorhandenen Spektralwerte möglichst grob zu quantisieren. Es wird daher angestrebt, einerseits hörbare Störungen im wieder decodierten Audiosignal zu vermeiden und andererseits möglichst wenig Bits zu verwenden, um das Audiosignal zu codieren bzw. hier zu quantisieren. Die durch die Quantisierung eingeführten Störungen, d. h. das Quantisierungsrauschen, soll unter der Maskierungsschwelle liegen und somit unhörbar sein. Gemäß bekannter Verfahren wird daher eine Einteilung der Spektralwerte in sog. Skalenfaktorbänder durchgeführt, die den Frequenzgruppen des menschlichen Ohrs entsprechen sollten. Spektralwerte in einer Skalenfaktorgruppe werden mit einem Skalenfaktor multipliziert, um Spektralwerte eines Skalenfaktorbandes insgesamt zu skalieren. Die durch den Skalenfaktor skalierten Skalenfaktorbänder werden anschließend quantisiert, woraufhin quantisierte Spektralwerte entstehen. Selbstverständlich ist eine Gruppierung in Skalenfaktorbänder nicht entscheidend. Sie wird jedoch bei den Standards MPEG-Layer 3 bzw. bei dem Standard MPEG-2 AAC (AAC = Advanced Audio Coding) verwendet.

Ein sehr wesentlicher Aspekt der Datenreduzierung besteht in der nach dem Quantisieren folgenden Entropie-Codierung der quantisierten Spektralwerte. Für die Entropiecodierung wird üblicherweise eine Huffman-Codierung verwendet. Unter einer Huffman-Codierung versteht man eine Codierung mit variabler Länge, d. h. die Länge des Codeworts für einen zu codierenden Wert ist abhängig von dessen Auftrittswahrscheinlichkeit. Logischerweise ordnet man dem wahrscheinlichsten Zeichen den kürzesten Code, d. h. das kürzeste Codewort, zu,

so daß mit der Huffman-Codierung eine sehr gute Redundanzreduktion erreicht werden kann. Ein Beispiel für eine allseits bekannte Codierung mit allgemeiner Länge ist das Morse-Alphabet.

In der Audiocodierung werden Huffman-Codes zur Codierung der quantisierten Spektralwerte benutzt. Ein moderner Audio-Coder, der beispielsweise nach dem Standard MPEG-2 AAC arbeitet, verwendet zur Codierung der quantisierten Spektralwerte verschiedene Huffman-Codetabellen, die dem Spektrum nach bestimmten Kriterien abschnittsweise zugeordnet werden. Dabei werden immer 2 oder 4 Spektralwerte in einem Codewort gemeinsam codiert.

Ein Unterschied des Verfahrens nach MPEG-2 AAC gegenüber dem Verfahren MPEG-Layer 3 besteht nun darin, daß verschiedene Skalenfaktorbänder, d. h. verschiedene Spektralwerte, zu beliebig vielen Spektralabschnitten oder "Sections" gruppiert werden. Bei AAC umfaßt ein Spektralabschnitt oder eine "Section" umfaßt zumindest vier Spektralwerte aber vorzugsweise mehr als vier Spektralwerte. Der gesamte Frequenzbereich der Spektralwerte wird daher in benachbarte Sections aufgeteilt, wobei eine Section ein Frequenzband darstellt, derart, daß alle Sections zusammen den gesamten Frequenzbereich, der durch die Spektralwerte nach der Transformation derselben überdeckt wird, umfassen.

Einem Abschnitt wird nun ebenso wie beim MPEG-Layer-3-Verfahren zum Erreichen einer maximalen Redundanzreduktion eine sog. Huffman Tabelle aus einer Mehrzahl derartiger Tabellen zugeordnet. Im Bitstrom des AAC-Verfahrens, welches üblicherweise 1024 Spektralwerte aufweist, befinden sich nun die Huffman-Codewörter für die Spektralwerte in aufsteigender Frequenzreihenfolge. Die Information über die in jedem Frequenzabschnitt verwendete Tabelle wird in den Seiteninformationen übertragen. Diese Situation ist in Fig. 6 dargestellt.

Fig. 6 stellt den beispielhaften Fall dar, bei dem der Bitstrom 10 Huffman-Codeworte umfaßt. Wenn immer aus einem Spektralwert ein Codewort gebildet wird, so können hier 10 Spektralwerte codiert sein. Üblicherweise werden jedoch immer 2 oder 4 Spektralwerte durch ein Codewort gemeinsam codiert, weshalb Fig. 6 einen Teil des codierten Bitstroms darstellt, der 20 bzw. 40 Spektralwerte umfaßt. In dem Fall, in dem jedes Huffman-Codewort 2 Spektralwerte umfaßt, stellt das mit der Nr. 1 bezeichnete Codewort die ersten 2 Spektralwerte dar, wobei die Länge des Codeworts Nr. 1 relativ klein ist, was bedeutet, daß die Werte der beiden ersten Spektralwerte, d. h. der beiden niedrigsten Frequenzkoeffizienten, relativ häufig auftreten. Das Codewort mit der Nr. 2 hingegen besitzt eine relativ große Länge, was bedeutet, daß die Beträge des 3. und 4. Spektralkoeffizienten im codierten Audiosignal relativ selten sind, weshalb dieselben mit einer relativ großen Bitmenge codiert werden. Aus Fig. 6 ist ferner ersichtlich, daß die Codewörter mit den Nr. 3, 4 und 5, die die Spektralkoeffizienten 5 und 6, bzw. 7 und 8 bzw. 9 und 10 darstellen, ebenfalls relativ häufig auftreten, da die Länge der einzelnen Codewörter relativ gering ist. Ähnliches gilt für die Codewörter mit den Nr. 6 - 10.

Wie es bereits erwähnt wurde, ist es aus Fig. 6 deutlich ersichtlich, daß die Huffman-Codewörter für die codierten Spektralwerte bezüglich der Frequenz linear ansteigend im Bitstrom angeordnet sind, wenn ein Bitstrom betrachtet wird, der durch eine bekannte Codiervorrichtung erzeugt wird.

Ein großer Nachteil von Huffman-Codes im Falle fehlerbehafteter Kanäle ist die Fehlerfortpflanzung. Es sei beispielsweise angenommen, daß das Codewort Nr. 2 in Fig. 6 gestört ist. Mit einer gewissen nicht niedrigen Wahrscheinlichkeit ist dann auch die Länge dieses falschen Codeworts Nr. 2 verändert. Dieselbe unterscheidet sich somit von der richtigen Länge. Wenn im Beispiel von Fig. 6 das Codewort Nr. 2

bezüglich seiner Länge durch eine Störung verändert worden ist, ist es für einen Codierer nicht mehr möglich, die Anfänge der Codewörter 3 - 10, d. h. fast des gesamten dargestellten Audiosignals, zu bestimmen. Es können also auch alle anderen Codewörter nach dem gestörten Codewort nicht mehr richtig decodiert werden, da nicht bekannt ist, wo diese Codewörter beginnen, und da ein falscher Startpunkt aufgrund des Fehlers gewählt wurde.

Das europäische Patent Nr. 0612156 schlägt als Lösung für das Problem der Fehlerfortpflanzung vor, einen Teil der Codewörter variabler Länge in einem Raster anzuordnen, und die restlichen Codewörter in die verbleibenden Lücken zu verteilen, so daß ohne vollständige Decodierung oder bei fehlerhafter Übertragung der Anfang eines im Raster angeordneten Codeworts leichter gefunden werden kann.

Das bekannte Verfahren schafft für die Fehlerfortpflanzung zwar eine teilweise Abhilfe durch Umsortierung der Codewörter. Für manche Codewörter wird ein fester Platz im Bitstrom vereinbart, während für die restlichen Codewörter die verbleibenden Zwischenräume zur Verfügung stehen. Dies kostet keine zusätzlichen Bits, verhindert aber im Fehlerfall die Fehlerfortpflanzung unter den umsortierten Codewörtern.

Die deutsche Patentanmeldung 19747119.6-31, die nach dem Anmeldetag der vorliegenden Anmeldung veröffentlicht wird, schlägt vor, nicht nur irgendwelche Codeworte an Rasterpunkten anzuordnen, sondern psychoakustisch bedeutsame Codeworte, d. h. Codeworte für Spektralwerte, die einen bedeutsamen Beitrag zum Audiosignal liefern, an Rasterpunkten anzuordnen. Ein Datenstrom mit Codeworten variabler Länge, wie er von einem solchen Codierer erzeugt wird, ist in Fig. 5 gezeigt. Der Datenstrom umfaßt ebenfalls wie in Fig. 6 10 Codeworte, wobei die Prioritätscodeworte schraffiert sind. Das erste Prioritätscodewort ist an einem ersten

Rasterpunkt 100 beginnend angeordnet, das zweite Prioritäts-codewort ist an einem zweiten Rasterpunkt 101 beginnend angeordnet, das dritte Prioritätscodewort ist an einem dritten Rasterpunkt 102 beginnend angeordnet, das vierte Prioritäts-codewort ist an einem vierten Rasterpunkt 103 beginnend angeordnet, und das fünfte Prioritätscodewort ist an einem fünften Rasterpunkt 104 beginnend angeordnet. Durch die Rasterpunkte 100 und 101 ist ein erstes Segment 105 definiert. Auf ähnliche Art und Weise ist ein zweites 106, ein drittes 107, ein viertes 108 und ein Abschlußsegment 109 definiert. In Fig. 5 ist gezeigt, daß die ersten beiden Segmente 105 und 106 eine andere Länge als die beiden Segmente 107 und 108 und wieder eine andere Länge als das Schlußsegment 109 haben. Die Nicht-Prioritätscodeworte 6, 7, 8, 9 und 10 werden an die Prioritätscodeworte anschließend in den Datenstrom eingetragen, derart, daß derselbe gewissermaßen aufgefüllt wird. Wie es in Fig. 5 gezeigt ist, werden bei dem nachveröffentlichten Verfahren die Nicht-Prioritätscodeworte fortlaufend in das Raster eingefügt, nachdem die Prioritätscodeworte geschrieben worden sind. Im einzelnen wird das Nicht-Prioritätscodewort Nr. 6 an das Nicht-Prioritätscodewort 1 anschließend eingetragen. Der dann noch in dem Segment 105 verbleibende Platz wird mit den anschließenden Nicht-Prioritätscodewort 7 aufgefüllt, wobei der Rest des Nicht-Prioritätscodeworts 7, d. h. 7b, in den nächsten freien Platz, d. h. in das Segment 107 direkt an das Prioritätscodewort 3 anschließend geschrieben wird. Entsprechend wird mit den Nicht-Prioritätscodeworten 8 bis 10 verfahren.

Das in Fig. 5 dargestellte nachveröffentlichte Verfahren hat den Vorteil, daß die Prioritätscodeworte 1 bis 5 vor einer Fehlerfortpflanzung geschützt sind, da ihre Anfangspunkte mit Rasterpunkten zusammenfallen und damit bekannt sind.

Ist nun beispielsweise das Prioritätscodewort 2 bei der Übertragung beschädigt worden, so wird bei dem in Fig. 6

gezeigten Stand der Technik ein Decodierer sehr wahrscheinlich keines der restlichen Codeworte 3 bis 10 mehr korrekt decodieren können. Bei dem in Fig. 5 gezeigten Verfahren fängt jedoch das nächste Codewort, d. h. das Prioritätscodewort 3, an dem Rasterpunkt 102 an, derart, daß der Decodierer auf jeden Fall den korrekten Anfang des Codeworts 3 finden wird. Somit wird bei dem in Fig. 5 gezeigten Verfahren überhaupt kein Folgefehler auftreten, und es wird nur das Prioritätscodewort Nr. 2 beschädigt sein. Dieses Verfahren liefert somit einen effektiven Schutz für Prioritätscodeworte, die a Rasterpunkten angeordnet sind.

Es besteht jedoch kein effektiver Schutz für Nicht-Prioritätscodeworte. Bezugnehmend auf Fig. 5 wird eine Beschädigung des Nicht-Prioritätscodeworts Nr. 6, derart, daß der Decodierer als falsches Codewort Nr. 6 eine um ein Bit kürzeres Codewort annimmt, dazu führen, daß auch das Codewort 7 nicht mehr korrekt decodiert werden kann, da das letzte Bit des korrekten Codeworts Nr. 6 bereits als Anfang des nächsten Codeworts Nr. 7 interpretiert wird. Somit wird ein Fehler im Codewort Nr. 6 dazu führen, daß sehr wahrscheinlich sämtliche daran anschließende Codeworte aufgrund eines Folgefehlers nicht mehr korrekt decodiert werden können, selbst wenn sie nicht durch einen Übertragungsfehler beeinträchtigt worden sind.

Die Aufgabe der vorliegenden Erfindung besteht darin, Datenströme aus Codeworten variabler Länge fehlerrobuster zu machen.

Diese Aufgabe wird durch eine Vorrichtung zum Erzeugen eines Datenstroms nach Patentanspruch 1, durch eine Vorrichtung zum Lesen eines Datenstroms nach Patentanspruch 12, durch ein Verfahren zum Erzeugen eines Datenstroms nach Patentanspruch 18 und durch ein Verfahren zum Lesen eines Datenstroms nach Patentanspruch 19 gelöst.

Der vorliegenden Erfindung liegt die Erkenntnis zugrunde, daß die Robustheit eines Datenstroms gegenüber Übertragungsfehlern, und insbesondere gegenüber Folgefehlern bei Codeworten variabler Länge entscheidend erhöht werden kann, wenn der Datenstrom nicht nur in einer Schreibrichtung geschrieben wird, sondern zusätzlich in der anderen Schreibrichtung. Im allgemeinsten Fall wird ein Datenstrom immer einen Anfang und ein Ende haben. Im Stand der Technik wurde der Datenstrom im einfachsten Fall ausgehend vom Anfangspunkt beschrieben, bis er zu Ende war. Dadurch konnte ein Übertragungsfehler im ersten Codewort dazu führen, daß der gesamte Datenstrom nicht mehr korrekt decodiert werden konnte, selbst wenn sämtliche anderen Codeworte korrekt übertragen wurden. Erfindungsgemäß könnte ein solcher Datenstrom derart geschrieben werden, daß die erste Hälfte des Datenstroms ausgehend vom Anfang des Datenstroms geschrieben wird, während die zweite Hälfte des Datenstroms ausgehend vom Ende des Datenstroms geschrieben wird. Bereits aus diesem einfachen Beispiel ist zu sehen, daß ein Übertragungsfehler in der ersten Hälfte des Datenstroms nicht mehr dazu führt, daß auch Codeworte des zweiten Datenstroms aufgrund von Folgefehlern nicht mehr korrekt decodiert werden können. Dies ist der Fall, da ein Decodierer weiß, daß er nach der Hälfte des Datenstroms vom Ende des Datenstroms aus weiter lesen muß, und zwar in der entgegengesetzten Leserichtung. Damit wird lediglich aufgrund der Umkehrung der Schreibrichtung/Lese- richtung praktisch ohne Mehraufwand eine gewisse Fehlerrobustheit erreicht.

Wie es bereits erwähnt worden ist, werden Codeworte mit variabler Länge unter Verwendung von Rasterpunkten in einen Datenstrom geschrieben, derart, daß ein Decodierer mit einer begrenzten Anzahl von Folgefehlern decodieren kann, da per Definition bestimmte Codeworte an Rasterpunkten beginnen. Zur maximalen Fehlerrobustheit wird prinzipiell ein möglichst enges Raster erwünscht sein, derart, daß ein Decodierer die korrekten Anfangspunkte möglichst vieler Code-

worte finden kann. Andererseits führt eine Erhöhung der Anzahl der Rasterpunkte, d. h. eine Verkleinerung der Segmentlänge, dazu, daß immer weniger Codeworte, die ja variable Längen haben, vollständig in das Raster passen, weshalb Vorkehrungen getroffen werden, daß die Endabschnitte derselben in andere Segmente geschrieben werden, um beim Decodieren korrekt erfaßt werden zu können. Dies führt zu einem steigenden Mehraufwand mit steigender Anzahl von Rasterpunkten bzw. mit kleiner werdender Segmentlänge.

Im Stand der Technik wurden Codeworte ausgehend von einem Rasterpunkt in nur einer einzigen Schreibrichtung geschrieben, wie es bezugnehmend auf die Fig. 5 und 6 erläutert worden ist. Erfindungsgemäß werden nun Codeworte ausgehend von Rasterpunkten auch in der umgekehrten Schreibrichtung geschrieben, wodurch die Anzahl der Codeworte, die an Rasterpunkten beginnend geschrieben werden können, im wesentlichen ohne zusätzlichen Mehraufwand im günstigsten Fall verdoppelt wird. Das Schreiben des Datenstroms mittels einer Einrichtung zum Schreiben in einer ersten Schreibrichtung von einem Bezugspunkt aus mittels einer zweiten Einrichtung zum Schreiben in einer zu der ersten Schreibrichtung entgegengesetzten zweiten Schreibrichtung von einem anderen Bezugspunkt aus wird es möglich, nicht nur "eine Seite" eines Bezugspunkts, sondern beide Seiten eines Bezugspunkts für die Fehlerrobustheit, d. h. für eine Sicherung gegen Fortpflanzungs- oder Folgefehler, auszunutzen. Je nach Ausführungsform kann beispielsweise jedes zweite Codewort in der gleichen Richtung geschrieben werden, und jedes verbleibende Codewort in der anderen Richtung. Andererseits können die Codeworte variabler Länge in verschiedene Sätze von Codeworten beispielsweise entsprechend ihrer Priorität, eingeteilt werden, derart, daß beispielsweise sämtliche Codeworte des ersten Satzes beginnend an Rasterpunkten in der ersten Schreibrichtung geschrieben werden und sämtliche Codeworte des zweiten Satzes beginnend an Rasterpunkten in der zweiten Schreibrichtung geschrieben werden können.

Darüberhinaus können Reste von Codeworten in derselben Schreibrichtung wie die Anfangsabschnitte der Codeworte geschrieben werden, oder aber auch in der umgekehrten Schreibrichtung. Selbstverständlich müssen Vorkehrungen getroffen werden, daß ein Decodierer, d. h. eine Vorrichtung zum Lesen des Datenstroms, immer genau im Bilde ist, welche Schreibrichtung beim Schreiben verwendet worden ist. Dies könnte entweder fest eingestellt werden, oder aber als Seiteninformationen zu dem Datenstrom aus Codeworten variabler Länge übertragen werden. Dasselbe trifft auch für die Segmentlängen zu, wobei die Segmentlänge entweder über dem gesamten Datenstrom gleich sein kann oder variieren kann, wobei die aktuelle Segmentlänge selbstverständlich ebenfalls in einem Decodierer fest eingestellt sein kann oder über Seiteninformationen zusammen mit den Codeworten variabler Länge übertragen werden kann.

Bevorzugte Ausführungsbeispiele der vorliegenden Erfindung werden nachfolgend bezugnehmend auf die beiliegenden Zeichnungen detailliert erläutert. Es zeigen:

Fig. 1 eine erfindungsgemäße Vorrichtung zum Schreiben eines fehlerrobusten Datenstroms;

Fig. 2 eine erfindungsgemäße Vorrichtung zum Lesen eines fehlerrobusten Datenstroms;

Fig. 3 ein Verfahrensablaufdiagramm des erfindungsgemäßen Verfahrens anhand von drei Sätzen von Codeworten variabler Länge;

Fig. 4 ein Verfahrensablaufdiagramm zur Veranschaulichung des erfindungsgemäßen Verfahrens zum Lesen eines Datenstroms, der gemäß Fig. 3 erzeugt worden ist;

Fig. 5 ein Datenstrom, der durch eine bekannte Vorrichtung

erzeugt wird, bei dem die Prioritätscodeworte einer Fehlerfortpflanzung ausgesetzt sind; und

Fig. 6 ein Datenstrom, bei dem eine Sortierung in Prioritätscodeworte und Nicht-Prioritätscodeworte durchgeführt ist.

Bevor auf Fig. 1 näher eingegangen wird, sei angemerkt, daß eine Codierung mit Codeworten variabler Länge in der Technik auch als Entropie-Codierung bezeichnet wird. Ein Vertreter der Entropie-Codierung ist die sogenannte Huffman-Codierung. Prinzipiell werden bei der Huffman-Codierung die zu codierenden Informationssymbole statistisch untersucht, um für die Informationssymbole, die häufiger auftreten, kürzere Codeworte zu bestimmen als für Informationssymbole, die weniger häufig auftreten. Bei einem vollständigen Huffman-Code sind sämtliche Codeworte abgeschlossene Enden oder Zweige eines Codebaums. Ein Huffman-Decodierer liest einen Datenstrom mit Huffman-Codeworten beispielsweise seriell ein und springt anschaulich gesprochen mit jedem Bit, das er zusätzlich einliest, zu einer Verzweigung des festgelegten Codebaums, bis er nach einer bestimmten Anzahl von Sprüngen, die der Anzahl von Bits des Codeworts, d. h. der Länge des Codeworts, entspricht, an einem Zweigende ankommt, das keine weitere Verzweigung aufweist, und somit ein Codewort ist. Dann weiß der Decodierer, daß mit dem nächsten Bit ein neues Codewort beginnt. Dieses Verfahren wird so oft wiederholt, bis der Datenstrom vollständig eingelesen ist. Mit jedem Mal, zu dem der Huffman-Codierer wieder an den Anfangspunkt, d. h. die Wurzel des Baumes zurückspringt, liegt an seinem Ausgangspunkt ein Codewort vor. Da die Länge der Codeworte implizit durch die Codeworte selbst bzw. durch den im Codierer und im Decodierer bekannten Codebaum gegeben ist, ist zu sehen, daß eine Störung im Datenstrom, die zu einer Umkehrung eines Bits führt, den Decodierer in dem Codebaum gewissermaßen irreführt, derart, daß er zu einem anderen Codewort, d. h. einem falschen Codewort, gelangt, das sehr

wahrscheinlich eine andere Länge hat als das richtige Codewort. In diesem Fall wird der Decodierer, wenn er an dem falschen Codewort angelangt ist, wieder zurückspringen und aufgrund der dann folgenden Bits wieder von Verzweigungspunkt zu Verzweigungspunkt in dem Codebaum laufen. Der Decodierer hat jedoch keine Möglichkeit, einen Folgefehler zu vermeiden, es sei denn, daß er zufällig wieder auf die "richtige Spur" kommt.

Daher müssen für eine fehlerrobuste Übertragung Fehlersicherungen unternommen werden, wie sie die vorliegende Erfindung vorsieht. Die Vorrichtung zum Erzeugen eines Datenstroms aus Codeworten variabler Länge gemäß der vorliegenden Erfindung könnte daher gewissermaßen als Sende- oder Ausgangsstufe eines Huffman-Codierers fungieren, während die Vorrichtung zum Lesen eines Datenstroms aus Codeworten variabler Länge als Empfangs- bzw. Eingangsstufe eines Huffman-Decodierers wirken könnte. Daraus ist jedoch zu sehen, daß die vorliegende Erfindung nicht nur auf Huffman-Codierer anwendbar ist, sondern auf jeglichen Code mit Codeworten variabler Länge, der für Folgefehler anfällig ist.

Fig. 1 zeigt eine erfindungsgemäße Vorrichtung 10 zum Erzeugen eines fehlerrobusten Datenstroms an einem Ausgang 12, wenn Codeworte variabler Länge an einem Eingang 14 in die Vorrichtung 10 eingegeben werden. Die Vorrichtung umfaßt eine erste Einrichtung 16 zum Schreiben in einer ersten Schreibrichtung von einem ersten Bezugspunkt aus und eine zweite Einrichtung 18 zum Schreiben in einer zweiten Schreibrichtung von einem zweiten Bezugspunkt aus. Je nach Komplexität der Vorrichtung 10 können die Codeworte variabler Länge beide an die beiden Einrichtungen 16 und 18 zum Schreiben angelegt werden, wie es in Figur 1 durch einen einfachen Verzweigungspunkt 20 und einen entsprechenden Kombinationspunkt 22 dargestellt ist. Die Auswahl, welche Codeworte in welcher Richtung geschrieben werden, bzw.

welche Abschnitte von Codeworten in welcher Richtung geschrieben werden, würde dann durch die Einrichtungen 16 und 18 durchgeführt werden. Alternativ könnte statt dem Knoten 20 ein Demultiplexer vorhanden sein, der bestimmte Codeworte, beispielsweise Codeworte eines Satzes von Codeworten, der ersten Einrichtung 16 zuführt und bestimmte Codeworte der zweiten Einrichtung 18 zuführt. In Analogie dazu würde der Kombinationspunkt 22 dann durch einen Multiplexer ausgeführt sein, der den fehlerrobusten Datenstrom 12 multiplext. Andere entsprechend gesteuerte Einrichtungen zum Speisen der beiden Einrichtungen 16 und 18 mit den Codeworten variabler Länge sind für Fachleute im Lichte der vorliegenden Beschreibung offensichtlich.

Eine zur Vorrichtung 10 zum Erzeugen eines Datenstroms, die in Fig. 1 gezeigt ist, komplementäre Vorrichtung 30 zum Lesen eines fehlerrobusten Datenstroms ist in Fig. 2 gezeigt. Dieselbe umfaßt einen Eingang 32, in dem der fehlerrobuste Datenstrom nach einer Übertragung beispielsweise über eine Funkstrecke, eingegeben wird um an einem Ausgangspunkt 34 wieder die Codeworte variabler Länge zu erhalten, die in den Eingang 14 der Vorrichtung 10 aus Fig. 1 eingespeist worden sind. Die Vorrichtung 30 zum Lesen des Datenstroms umfaßt eine erste Einrichtung 36 zum Lesen in der ersten Richtung von dem ersten Bezugspunkt aus und eine zweite Einrichtung 38 zum Lesen des Datenstroms in der zweiten Richtung von einem zweiten Bezugspunkt aus.

Die Vorrichtung 30 enthält selbstverständlich ebenfalls einen Verzweigungspunkt 40 und einen Kombinationspunkt 42, wobei die Einspeisung des fehlerrobusten Datenstroms in die beiden Einrichtungen 36 und 38 beispielsweise basierend auf einem festeingestellten Algorithmus stattfindet oder basierend auf Seiteninformationen, die zusammen mit dem fehlerrobusten Datenstrom ebenfalls von dem Sender d. h. der Vorrichtungen 10 in Fig. 1, zu dem Empfänger, d. h. der Vorrichtung 30 in Fig. 2 übertragen werden können.

In Fig. 3 ist anhand eines Beispiels das erfindungsgemäße Verfahren zum Schreiben von Codeworten variabler Länge dargestellt. Im Beispiel existieren 15 Codeworte variabler Länge 30, die vorzugsweise in einen ersten Satz mit 6 Codeworten 1 bis 6, in einen zweiten Satz mit ebenfalls 6 Codeworten 7 bis 12 und in einen dritten Satz mit den restlichen 3 Codeworten 13 bis 15 aufgeteilt sind. Wie es in Fig. 3 gezeigt ist, haben die Codeworte 30 variable Längen.

Gemäß einem bevorzugten Ausführungsbeispiel der vorliegenden Erfindung ist die Segmentlänge, d. h. die Länge eines Segments, größer als die Länge des längsten Codeworts des ersten Satzes. Die Codeworte des ersten Satzes werden an Rasterpunkten 41 bis 46 angeordnet, wobei für das letzte Segment Nr. 6 ein Rasterpunkt gestrichelt angedeutet ist, der jedoch nicht verwendet wird, da das Ende 47 des Datenstroms gewissermaßen ebenfalls als Rasterpunkt angesehen werden kann und der gestrichelt angedeutete Rasterpunkt somit überflüssig ist. Das erste Segment Nr. 6 ist daher länger als die anderen Segmente, was für die vorliegende Erfindung jedoch völlig unerheblich ist. Die Segmente können allgemein gesagt beliebige Längen haben, die sich innerhalb des Datenstroms ändern, wobei jedoch die aktuelle Länge eines Segments selbstverständlich dem Decodierer bekannt sein muß, damit die erfindungsgemäßen Vorteile genutzt werden können.

Zunächst werden in einem Schritt a) die Codeworte des ersten Satzes in den Datenstrom geschrieben, woraus sich ein mit 31 bezeichneter bruchstückhafter Datenstrom ergibt, bei dem die Codeworte des ersten Satzes von links nach rechts in ein jeweiliges Segment geschrieben werden, wie es durch Pfeile 48 angedeutet ist, die in der gesamten Fig. 3 die Schreibrichtung symbolisieren sollen. Da die Segmentlänge länger als die größte Länge eines Codeworts des ersten Satzes gewählt ist, wird für den Schritt a) lediglich ein einziger

Versuch benötigt. Ist die Segmentlänge kürzer, können entsprechend mehr Versuche erforderlich sein.

In einem Schritt b) werden nun die Codeworte des zweiten Satzes in den Datenstrom 31 geschrieben. Um eine hohe Fehlerrobustheit zu erreichen, werden die Codeworte des zweiten Satzes nicht von links nach rechts wie die Codeworte des ersten Satzes geschrieben, sondern jeweils ausgehend von dem zweiten Rasterpunkt z.B. dem Rasterpunkt 42 für das erste Segment, von rechts nach links geschrieben, wie es durch den entsprechenden Schreibrichtungspfeil angedeutet ist. Das Schreiben der Codeworte des zweiten Satzes findet nach einer vorbestimmten Zuordnungsvorschrift statt, die bei dem gewählten Beispiel derart lautet, daß das erste Codewort des zweiten Satzes in dem gleichen Segment geschrieben werden soll wie das erste Codewort des ersten Satzes, jedoch immer unter der Voraussetzung, daß in diesem Segment noch Platz ist. Der aus dem ersten Versuch entstandene Datenstrom 32 zeigt, daß im ersten Segment lediglich soviel Platz war, den Anfangsabschnitt des Codeworts Nr. 7 zu schreiben.

Im Gegensatz zum Stand der Technik, bei dem der zweite Teil des Codeworts Nr. 7 in das zweite Segment geschrieben worden wäre, wird die zweite Hälfte des Codeworts Nr. 7, d. h. 7b, gespeichert, um dasselbe nach einer vorbestimmten Vorschrift, d. h. nach einer Vorschrift die auch dem Decodierer bekannt sein muß, in einem zweiten Versuch in den Datenstrom zu schreiben. Fig. 3 macht deutlich, daß in dem zweiten Segment zwischen dem Codewort Nr. 2 und 8 noch soviel Platz vorhanden war, daß der Endabschnitt des Codeworts Nr. 7 eingetragen werden konnte. Wäre nicht genug Platz gewesen, so wäre der dritte Abschnitt des Codeworts in das Segment Nr. 3 eingetragen worden. Die vorbestimmte Vorschrift zum Eintragen des Codeworts Nr. 7 in den Datenstrom besteht bei Fig. 3 also darin, immer um ein Segment weiter zu gehen. Selbstverständlich könnte auch immer um zwei Segmente weitergegangen werden, oder um drei oder mehr, derart, daß

dann der zweite Abschnitt 7b statt in das zweite Segment in das dritte, im nächsten Versuch in das fünfte etc. geschrieben werden könnte. Die Reihenfolge der Segmente, die verwendet wird, um den zweiten Teil des Abschnitt 7 irgendwo unterzubringen, ist beliebig. Sie muß jedoch dem Decodierer transparent sein, damit der umsortierte Datenstrom wieder gelesen werden kann.

In den entstandenen ebenfalls noch bruchstückhaften Datenstrom 33 sollen nun die Codeworte des dritten Satzes 13 bis 15 eingetragen werden. In Analogie zu dem Schritt b) geschieht dies vorzugsweise anhand der selben Zuordnungsvorschrift, derart, daß das erste Codewort des dritten Satzes dem ersten Segment zugeordnet ist, das zweite Codewort des dritten Satzes dem zweiten Segment zugeordnet ist, das dritte Codewort des dritten Satzes dem dritten Segment zugeordnet ist usw. Diese Zuordnungsvorschrift ist für den dritten Satz völlig beliebig und kann sich auch von der Zuordnungsvorschrift für den zweiten Satz unterscheiden, wobei erfindungsgemäß jedes Codewort eines Satzes einem anderen Segment zugeordnet ist. Genauso ist die Schreibrichtung für jeden Satz beliebig wählbar. Vorzugsweise wird eine alternierende Schreibrichtungsreihenfolge verwendet. Alternativ können jedoch auch zwei benachbarte Sätze mit der gleichen Schreibrichtung geschrieben werden. Prinzipiell kann auch die Schreibrichtung innerhalb eines Satzes verändert werden.

Der erste Versuch im Schritt c) war lediglich darin erfolgreich, den ersten Abschnitt des Codeworts Nr. 15 einzutragen, wodurch sich ein bruchstückhafter Datenstrom 34 ergeben hat. Die Codeworte 13, 14 und der zweite Abschnitt des Codeworts 15, d. h. 15b, werden gespeichert, um im zweiten, dritten, vierten, fünften und sechsten Versuch untergebracht zu werden, wobei im zweiten Versuch der zweite Abschnitt 15b im vierten Segment untergebracht werden konnte (Datenstrom 35), im dritten Versuch nichts untergebracht werden konnte,

im vierten Versuch der Anfangsabschnitt des Codeworts 14 untergebracht werden konnte (Datenstrom 36), im fünften Versuch der Endabschnitt des Codeworts 14, d. h. 14b, untergebracht werden konnte (Datenstrom 37), und schließlich im sechsten und letzten Versuch das erste Codewort des dritten Satzes im sechsten Segment eingetragen werden konnte, wodurch sich der fehlerrobuste Datenstrom 38 für das hier skizzierte Beispiel ergibt. Das anhand von Fig. 3 beschriebene Verfahren stellt sicher, daß die Länge des fehlerrobusten Datenstroms genau der Summe der Längen der Codeworte variabler Länge entspricht, was im Sinne einer Entropiecodierung zur Datenreduzierung selbstverständlich ist. Die vorliegende Erfindung ist jedoch nicht darauf begrenzt, daß der fehlerrobuste Datenstrom die minimale Länge hat, da die Fehlerrobustheit nicht durch unter Umständen vorhandene Leerbits beeinträchtigt wird.

Wenn der in Fig. 3 gegebene robuste Datenstrom betrachtet wird, so ist zu sehen, daß der Anfang des Codeworts Nr. 8, d. h. der Rasterpunkt 43, völlig unabhängig von dem Ende des Codeworts Nr. 7 ist. Auch der Anfang des Codeworts Nr. 9, d. h. der Rasterpunkt 44, ist völlig unabhängig von dem Ende des Codeworts Nr. 8. Ferner sei darauf hingewiesen, daß aufgrund der entgegengesetzten Schreibreihenfolge beispielsweise ein Datenfehler im Codewort Nr. 1 im ersten Segment, der dazu führt, daß das falsche Codewort aufgrund des Datenfehlers um ein Bit kürzer als das korrekte Codewort Nr. 1 ist, nicht zu einer Zerstörung des Anfangsabschnitts des Codeworts Nr. 7a führt, da dasselbe statt von links nach rechts von rechts nach links geschrieben wurde. Wäre dasselbe von links nach rechts geschrieben worden, so würde ein Decodierer das noch übrige Bit von dem ursprünglich korrekten Codewort Nr. 1 als Anfangsbit des Codeworts Nr. 7 nehmen, wodurch sich ein Folgefehler von 1 auf 7 ergeben würde. Dieser Folgefehler würde sich jedoch nicht auf 8 fortpflanzen, da Codewort Nr. 8 wieder völlig unabhängig von Codewort Nr. 7 ist, da die Schreibreihenfolge von rechts nach links

gewählt wurde. Ist die Schreibreihenfolge des Codeworts Nr. 8 gleich der Schreibreihenfolge der Codeworte des ersten Satzes, so würde sich der Fehler ebenfalls nicht von 7 auf 8 fortpflanzen, da das Codewort Nr. 8 vor dem zweiten Teil 7b aufgrund der Zuordnungsvorschrift an das Codewort Nr. 2 angrenzend geschrieben werden würde und somit nicht durch einen falschen Abschnitt 7b beeinflusst wird.

Fig. 4 zeigt anhand eines entsprechenden Beispiels die Funktionsweise der Vorrichtung zum Lesen des fehlerrobusten Datenstroms 38 auf. Zunächst werden in einem Schritt a) die Codeworte des ersten Satzes aus dem fehlerrobusten Datenstroms extrahiert. Dazu liest die erfindungsgemäße Vorrichtung, die mit einem Huffman-Decodierer gekoppelt sein dürfte, ausgehend von dem ersten Rasterpunkt 41 das Codewort des ersten Satzes, ausgehend von dem zweiten Rasterpunkt 42 das Codeworts Nr. 2 des ersten Satzes, usw. bis alle Codeworte 1 bis 6 des ersten Satzes eingelesen sind. Selbstverständlich wählt auch die Vorrichtung zum Lesen des Datenstroms die selbe Richtung, wie sie von der Vorrichtung zum Erzeugen angewendet worden ist.

Anschließend werden in einem Schritt b) aus dem noch verbleibenden Datenstrom 50 die Codeworte des zweiten Satzes extrahiert. Hierbei springt der Decodierer an den zweiten Rasterpunkt 42 des ersten Segments und erhält den Anfangsabschnitt des Codeworts 7 des zweiten Satzes (das erste Segment ist jetzt leer) und liest dann nicht den zweiten Abschnitt 7b ein, sondern 7a wird zunächst gespeichert, um dann das zweite Codewort des zweiten Satzes ausgehend von dem zweiten Rasterpunkt des zweiten Segments usw. einzulesen. Es ergibt sich somit ein Restdatenstrom 51, in dem das erste Segment schon vollständig geleert ist. Da der Decodierer nun nicht das Codewort 7 durchgehend liest sondern immer segmentweise aufgrund der Vorrichtung zum Erzeugen des Datenstroms verwendeten Zuordnungsvorschrift liest, wird die bereits beschriebene Fehlerrobustheit si-

chergestellt, die eine Ausbreitung von Folgefehlern stark reduziert.

In einem zweiten Versuch zum Extrahieren der Codeworte des zweiten Satzes wird nun im zweiten Segment entsprechend der vorhandenen Schreibrichtung der zweite Teil des Codeworts 7b gelesen, woraufhin in dem resultierenden Datenstrom 52 nur noch Codeworte des dritten Satzes verbleiben, und das zweite Segment leer ist. Diese werden in einem Schritt c) extrahiert, wobei zunächst in einem ersten Versuch der Anfangsabschnitt des Codeworts 15 eruiert worden ist, der jedoch gespeichert wird, da das Codewort 15 nicht vollständig in dem dritten Segment aufgefunden worden ist. Das dritte Segment ist nun leer. In einem zweiten Versuch kann das Codewort 15 vollständig gefunden werden. Die Suche nach dem Codewort 14 im Segment 3 und nach dem Codewort 15 im Segment 4 blieb jedoch erfolglos, was durch den Datenstrom 54 sichtbar ist. Im vierten Versuch führte jedoch die Suche von Codewort 14 im fünften Segment zu einem positiven Ergebnis. Das Codewort 14 war jedoch nicht vollständig, weshalb der Anfangsabschnitt 14a gespeichert wurde, um dann in einem fünften Versuch den noch verbleibenden Datenstrom 55 zu untersuchen und schließlich in einem letzten sechsten Versuch den Datenstrom 56, der nur noch aus dem sechsten Segment und aus dem Codewort 13 besteht, vollständig einzulesen.

Obwohl im vorhergehenden Beispiel lediglich eine Stückelung in Anfangsabschnitt und Endabschnitt von Codeworten beispielhaft dargestellt worden ist, ist prinzipiell eine beliebige Stückelung möglich. Solange der Decodierer die Zuordnung von Codeworten des zweiten Satzes bzw. des dritten Satzes und weiterer Sätze zu jeweils unterschiedlichen Segmenten beachtet, wird eine fehlerrobuste Decodierung sichergestellt sein. Es ist ferner offensichtlich, daß die Einsortierung der Endabschnitte von Codeworten in den Datenstrom beliebig ist, solange der Decodierer bzw. die dem Decodierer vorgeschaltete Einleseschaltung genau weiß, welche

vorbestimmte Vorschrift im Codierer ausgeführt worden ist.

Um noch einmal die Vorteile bzw. die Funktionsweise der vorliegenden Erfindung herauszustellen, wird auf den fehlerrobusten Datenstrom Nr. 38 von Fig. 3 verwiesen. Wenn das erste Segment zwischen den Rasterpunkten 41 und 42 betrachtet wird, so ist zu sehen, daß das Codewort Nr. 1 ausgehend von dem ersten Rasterpunkt 41 von links nach rechts geschrieben wird, wie es durch den darunter gezeichneten Pfeil deutlich ist. Der erste Teil des Codeworts Nr. 7, d. h. 7a, wird dagegen ausgehend von dem zweiten Rasterpunkt 42 von rechts nach links geschrieben. Würden beide Codeworte Nr. 1 und 7 bzw. 7a nur von links nach rechts in den Datenstrom geschrieben werden, so würde der Anfang des Codewortes 7 bzw. der Anfangspunkt des Anfangsabschnitts 7a des Codeworts 7 von dem Endpunkt des Codeworts 1 abhängen. Ein Übertragungsfehler im Codewort 1 würde damit nahezu unweigerlich auch zu einem Folgefehler im Codewort 7 führen. Wird das Codewort 7 dagegen erfindungsgemäß ausgehend von dem zweiten Rasterpunkt 42 in umgekehrter Schreibrichtung geschrieben, so hängt der Anfangspunkt des Codeworts 7 bzw. des Anfangsabschnitts 7a des Codewortes 7 nicht mehr von dem Codewort 1 ab, sondern ist durch das Raster bzw. den Rasterpunkt 42 bestimmt. Ein Decodierer wird diesen Anfangspunkt immer wissen, weshalb ein Fehler im Codewort 1 nicht zu einem Fehler im Codewort 7 führen wird. Aus dem fehlerrobusten Datenstrom 38 von Fig. 3 ist zu sehen, daß der erste Abschnitt 7a und der zweite Abschnitt 7b des Codeworts Nr. 7 beide in derselben Schreibrichtung geschrieben werden. Dies ist jedoch nicht zwingend erforderlich. Selbstverständlich könnte der zweite Abschnitt 7b des Codeworts 7 auch von links nach rechts geschrieben werden und würde dann am Ende des zweiten Codewortes Nr. 2 beginnen.

Werden die Rasterpunkte derart gewählt, daß die Segmentlängen länger sind als die höchste Länge eines Codeworts des ersten Satzes, so wird kein Segment durch das Codewort des

ersten Satzes vollständig ausgefüllt, wie es beispielsweise aus dem Datenstrom 31 von Fig. 3 zu sehen ist. In diesem Fall wird die Anzahl der Codeworte, die an Rasterpunkten beginnend geschrieben werden können, tatsächlich verdoppelt, ohne einen einzigen zusätzlichen Rasterpunkt vorsehen zu müssen.

Patentansprüche

1. Vorrichtung (10) zum Erzeugen eines Datenstroms (38), der zwei Bezugspunkte (41 - 47) aufweist, aus Codeworten unterschiedlicher Länge, mit folgenden Merkmalen:

einer ersten Einrichtung (16) zum Schreiben zumindest eines Teils eines Codeworts in den Datenstrom in einer ersten Schreibrichtung von einem ersten Bezugspunkt aus;

einer zweiten Einrichtung (18) zum Schreiben zumindest eines Teils eines Codeworts in den Datenstrom in einer der ersten Schreibrichtung entgegengesetzten zweiten Schreibrichtung von dem anderen Bezugspunkt aus.

2. Vorrichtung Anspruch 1, bei der die zwei Bezugspunkte des Datenstroms der Anfang (41) bzw. das Ende (47) des Datenstroms sind.
3. Vorrichtung Anspruch 1, bei der die erste Schreibeinrichtung (16) angeordnet ist, um einen Anfangsabschnitt eines Codeworts zu schreiben, und bei der die zweite Schreibeinrichtung (18) angeordnet ist, und zumindest einen Teil des Rests des selben Codeworts zu schreiben.
4. Vorrichtung Anspruch 1, bei der der Datenstrom eine Vielzahl von Rasterpunkten (41 - 46) als Bezugspunkte aufweist, wobei die Rasterpunkte ein Raster festlegen, wobei zwei benachbarte Rasterpunkte ein Segment definieren, wobei die erste Schreibeinrichtung (16) angeordnet ist, um ein erstes Codewort zu schreiben, das an einem ersten Rasterpunkt eines Segments beginnt, und wobei die zweite Schreibeinrichtung (18) angeordnet ist, um ein zweites Codewort zu schreiben, das an dem zweiten Bezugspunkt des Segments beginnt.
5. Vorrichtung nach Anspruch 1, bei der der Datenstrom eine

Vielzahl von Rasterpunkten als Bezugspunkte aufweist, wobei die Rasterpunkte ein Raster festlegen, wobei zwei benachbarte Rasterpunkte ein Segment definieren, wobei die erste Schreibeinrichtung (16) angeordnet ist, um einen Anfangsabschnitt eines ersten Codewortes an einem ersten Rasterpunkt eines ersten Segments zu schreiben, und einen Anfangsabschnitt eines zweiten Codeworts an einem ersten Rasterpunkt eines folgenden Segments zu schreiben, und wobei die zweite Schreibeinrichtung (18) angeordnet ist, um den Rest des ersten Codeworts von dem zweiten Bezugspunkt des zweiten Segments aus bzw. den Rest des zweiten Codewortes von dem zweiten Bezugspunkt eines weiteren Segments aus zu schreiben.

6. Vorrichtung nach einem der vorhergehenden Ansprüche, bei der die Codeworte in eine Mehrzahl von Sätzen von Codeworten aufgeteilt sind, wobei die erste Einrichtung (16) zum Schreiben angeordnet ist, um jedes Codewort des ersten Satzes ausgehend von einem ersten Bezugspunkt eines Segments in den Datenstrom zu schreiben, und wobei die zweite Schreibeinrichtung (18) angeordnet ist, um jedes Codewort des zweiten Satzes ausgehend von einem zweiten Bezugspunkt eines jeweiligen Segments in den Datenstrom zu schreiben.
7. Vorrichtung nach Anspruch 6, bei der die zweite Schreibeinrichtung (18) angeordnet ist, um ausgehend von dem zweiten Bezugspunkt des Segments, in das die erste Schreibeinrichtung (16) geschrieben hat, zu schreiben, wobei, falls das entsprechende Codewort des zweiten Satzes länger als ein freier Platz in dem Segment ist, der Teil des Codewort des zweiten Satzes, der in den freien Platz paßt, in das Segment geschrieben wird, und der Rest gespeichert wird.
8. Vorrichtung nach Anspruch 6 oder 7, bei der die zweite Schreibeinrichtung (18) angeordnet ist, um erst dann ak-

tiv zu werden, wenn sämtliche Codewörter des ersten Satzes in den Datenstrom geschrieben sind.

9. Vorrichtung nach Anspruch 8, bei der die zweite Schreibeinrichtung (18) angeordnet ist, um den Rest des Codeworts an das Ende eines Codeworts des zweiten Satzes als Bezugspunkt in den Datenstrom zu schreiben.
10. Vorrichtung nach einem der Ansprüche 6 bis 9, bei der die Codeworte des zweiten Satzes nach einer vorbestimmten Zuordnungsvorschrift Segmenten zugeordnet sind, derart, daß jedes Codewort des zweiten Satzes einem unterschiedlichen Segment zugeordnet ist, und bei dem die zweite Schreibeinrichtung (18) angeordnet ist, um für den Fall, daß ein Codewort des zweiten Satzes nicht mehr in das zugeordnete Segment paßt, den Rest in ein nicht vollbesetztes anderes Segment zu schreiben, nachdem sie alle verbleibenden Segmente mit den anderen Codeworten des zweiten Satzes abgearbeitet hat.
11. Vorrichtung nach einem der Ansprüche 4 bis 10, bei der die Codeworte in zumindest drei Sätze unterteilt sind, wobei die erste Schreibeinrichtung (16) angeordnet ist, um die Codeworte des ersten Satzes ausgehend von ersten Rasterpunkten von Segmenten zu schreiben, bei der die zweite Schreibeinrichtung (18) angeordnet ist, um die Codeworte des zweiten Satzes ausgehend von den anderen Rasterpunkten der Segmente zu schreiben, und bei der die erste oder zweite Schreibeinrichtung (16, 18) ferner angeordnet ist, um den dritten Satz ausgehend von Enden der Codeworte des ersten bzw. zweiten Satzes zu schreiben.
12. Vorrichtung (30) zum Lesen eines Datenstroms (32), der zwei Bezugspunkte aufweist, von denen aus in einer ersten bzw. zweiten Schreibrichtung zumindest ein Teil eines Codeworts geschrieben ist, mit folgenden Merkma-

len:

einer ersten Einrichtung (36) zum Lesen von dem ersten Bezugspunkt aus in einer ersten Leserichtung, die der ersten Schreibrichtung entspricht; und

einer zweiten Einrichtung (38) zum Lesen von dem zweiten Bezugspunkt aus in einer zur ersten Leserichtung entgegengesetzten zweiten Leserichtung.

13. Vorrichtung nach Anspruch 12, bei der der Datenstrom eine Vielzahl von Rasterpunkten als Bezugspunkte aufweist, wobei die Rasterpunkte ein Raster festlegen, wobei zwei benachbarte Rasterpunkte ein Segment definieren, bei der der Datenstrom eine Mehrzahl von Sätzen von Codeworten aufweist, wobei ein erster Satz von Codeworten in der ersten Richtung geschrieben ist und ein zweiter Satz von Codeworten in einer zweiten Richtung geschrieben ist, wobei die Codeworte des zweiten Satzes Segmenten des Datenstroms nach einer vorbestimmten Zuordnungsvorschrift zugeordnet sind, derart, daß jedes Codewort eines Satzes einem anderen Segment zugeordnet ist, wobei die Vorrichtung ferner folgendes Merkmal aufweist:

eine Steuerungseinrichtung (40) zum Liefern der Codeworte des ersten Satzes zu der ersten Schreibeinrichtung (36), wobei das Codewort des ersten Satzes an einem Rasterpunkt beginnt, und zum Liefern der Codeworte des zweiten Satzes zu der zweiten Leseinrichtung (38), wobei gemäß der vorbestimmten Zuordnungsvorschrift zu entsprechenden Rasterpunkten gesprungen wird, wobei, falls an einem Bezugspunkt kein Codewort angetroffen wird, überprüft wird, ob an entsprechenden Rasterpunkten nach der Zuordnungsvorschrift Codeworte des zweiten Satzes vorhanden sind, und wobei, nachdem sämtliche Codeworte des zweiten Satzes gelesen sind, zu einem anderen Ra-

sterpunkt nach der vorbestimmten Vorschrift gesprungen wird, um sämtliche Codeworte des zweiten Satzes vollständig zu erhalten.

14. Vorrichtung nach Anspruch 13, bei der, wenn nur ein Anfangsabschnitt eines Codeworts durch eine Schreibeinrichtung in einem Segment gelesen wird, dieser Anfangsabschnitt gespeichert wird.
15. Vorrichtung nach einem der vorhergehenden Ansprüche, bei der die Codeworte Huffman-Codeworte sind.
16. Vorrichtung nach einem der vorhergehenden Ansprüche, bei der die Codeworte Informationssymbole darstellen, und Codeworte des ersten Satzes bedeutsamere Informationssymbole darstellen als Codeworte des zweiten Satzes oder weiterer Sätze.
17. Vorrichtung nach Anspruch 16, bei der die Informationssymbole Spektralwerte eines Audiosignals sind, und die Codeworte des ersten Satzes psychoakustisch bedeutsame Spektralwerte darstellen, die vor einer Fehlerfortpflanzung aufgrund eines Übertragungsfehlers in dem Datenstrom zu schützen sind.
18. Verfahren (10) zum Erzeugen eines Datenstroms (38), der zwei Bezugspunkte (41) bis (47) aufweist, aus Codeworten unterschiedlicher Länge, mit folgenden Schritten:

Schreiben zumindest eines Teils eines Codeworts in den Datenstrom in einer ersten Schreibrichtung von einem ersten Bezugspunkt aus;

Schreiben zumindest eines Teils eines Codeworts in den Datenstrom in einer der ersten Schreibrichtung entgegengesetzten zweiten Schreibrichtung von dem anderen Bezugspunkt aus.

19. Verfahren (30) zum Lesen eines Datenstroms (32), der zwei Bezugspunkte aufweist, von denen aus in einer ersten bzw. zweiten Schreibrichtung zumindest ein Teil eines Codeworts geschrieben ist, mit folgenden Schritten:

Lesen von dem ersten Bezugspunkt aus in einer ersten Leserichtung, die der ersten Schreibrichtung entspricht; und

Lesen von dem zweiten Bezugspunkt aus in einer zur ersten Leserichtung entgegengesetzten zweiten Leserichtung.

Vorrichtung und Verfahren zum Erzeugen eines Datenstroms und Vorrichtung und Verfahren zum Lesen eines Datenstroms

Zusammenfassung

Ein Entropiecodierer umfaßt eine Vorrichtung zum Erzeugen eines Datenstroms, der zwei Bezugspunkte aufweist aus Codeworten unterschiedlicher Länge, wobei die Vorrichtung eine erste Einrichtung zum Schreiben zumindest eines Teils eines Codewortes in den Datenstrom in einer ersten Schreibrichtung von einem ersten Bezugspunkt aus und eine zweite Einrichtung zum Schreiben zumindest eines Teils eines Codeworts in dem Datenstrom in einer der ersten Schreibrichtungen entgegengesetzten zweiten Schreibrichtung von dem anderen Bezugspunkt aus aufweist. Insbesondere wenn ein Raster mit einer Mehrzahl von Segmenten verwendet wird, um die Codeworte variabler Länge in den Datenstrom zu schreiben, wird die Anzahl der Codeworte, die an Rasterpunkten beginnend geschrieben werden können, im günstigsten Fall verdoppelt, derart, daß der Datenstrom aus Codeworten variabler Länge gegenüber einer Ausbreitung von Folgefehlern robust ist.

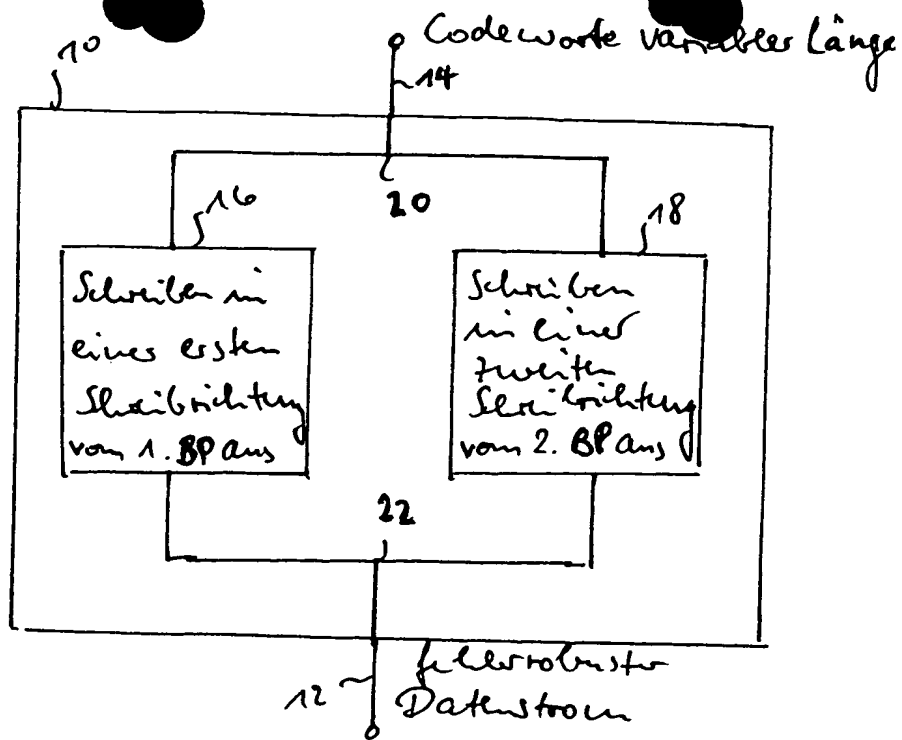


Fig. 1

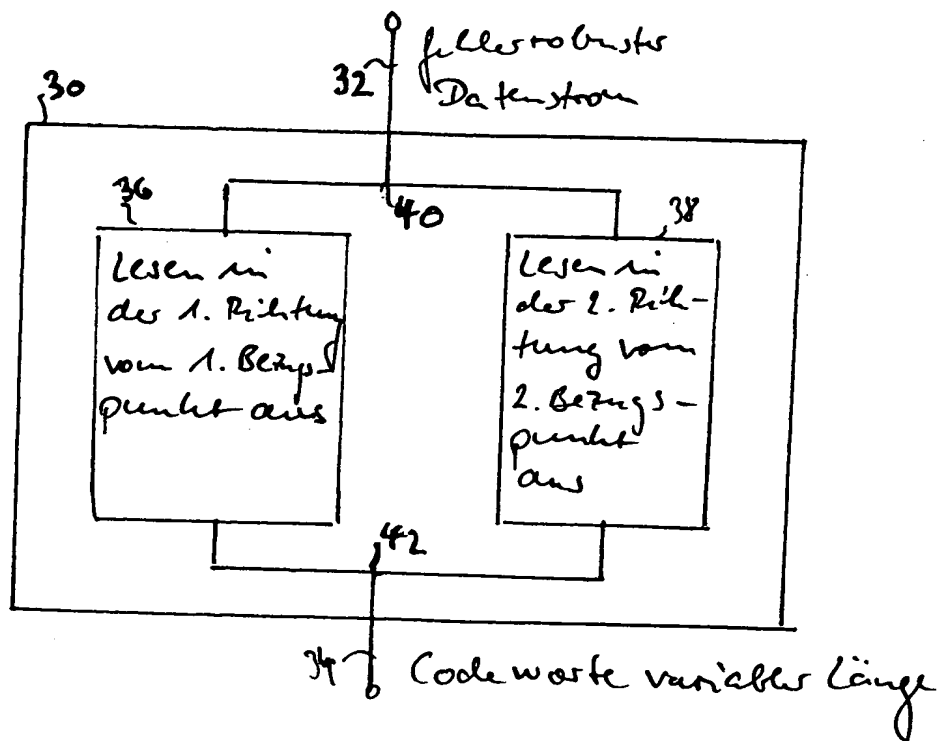
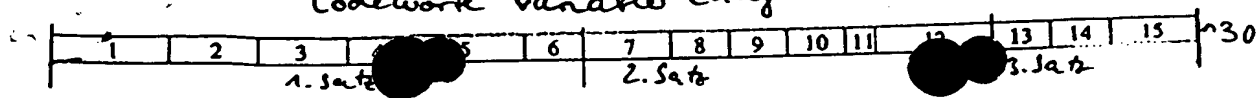
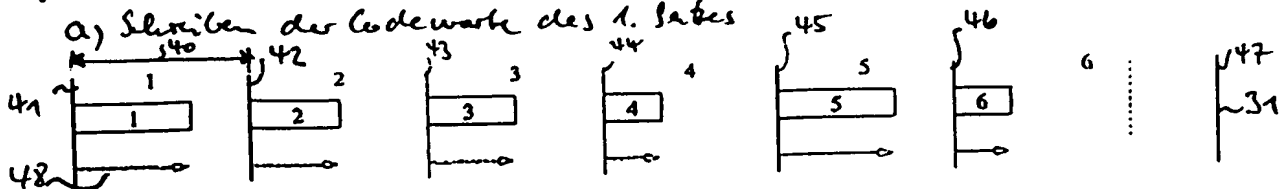


Fig. 2

Codewort variabler Länge

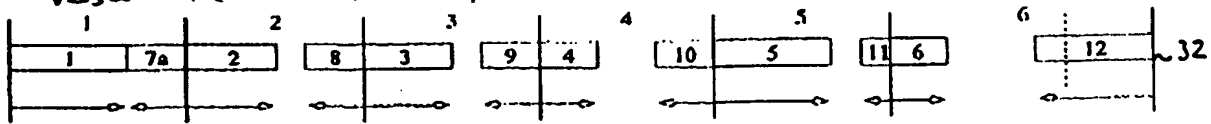


a) Schreiben der Codeworte des 1. Satzes

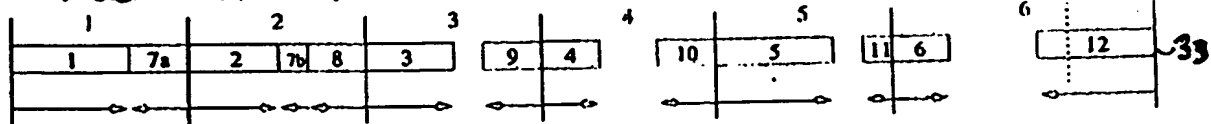


b) Schreiben der Codeworte des 2. Satzes

Versuch 1 (7 in 1, 8 in 2, 9 in 3, 10 in 4, 11 in 5, 12 in 6): Speichern von 7b

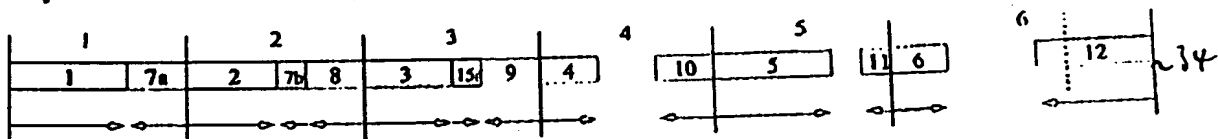


Versuch 2 (7 in 2)

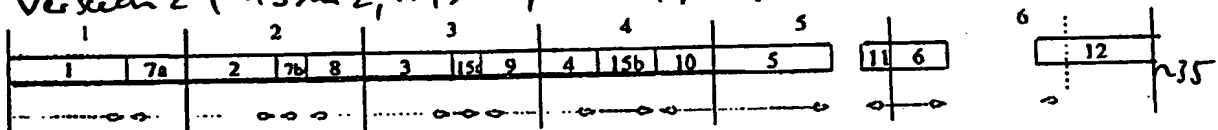


Schreiben der Codeworte des 3. Satzes

Versuch 1 (13 in 1, 14 in 2, 15 in 3): Speichern von 13, 14, 15b

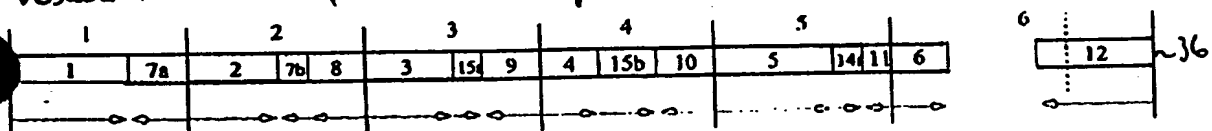


Versuch 2 (13 in 2, 14 in 3, 15 in 4): Speichern von 13, 14

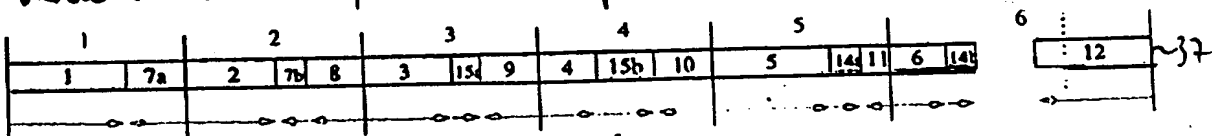


Versuch 3 (13 in 3, 14 in 4): Speichern von 13, 14

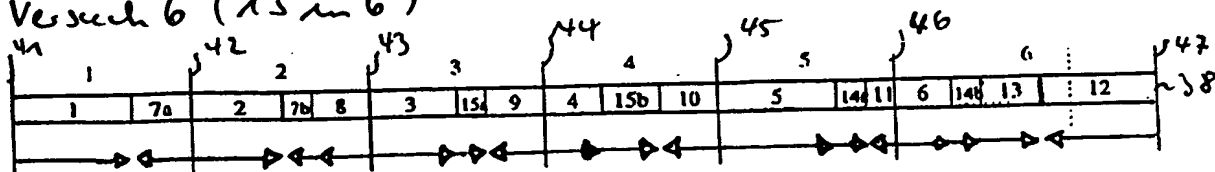
Versuch 4 (13 in 4, 14 in 5): Speichern von 13, 14b



Versuch 5 (13 in 5, 14 in 6): Speichern von 13



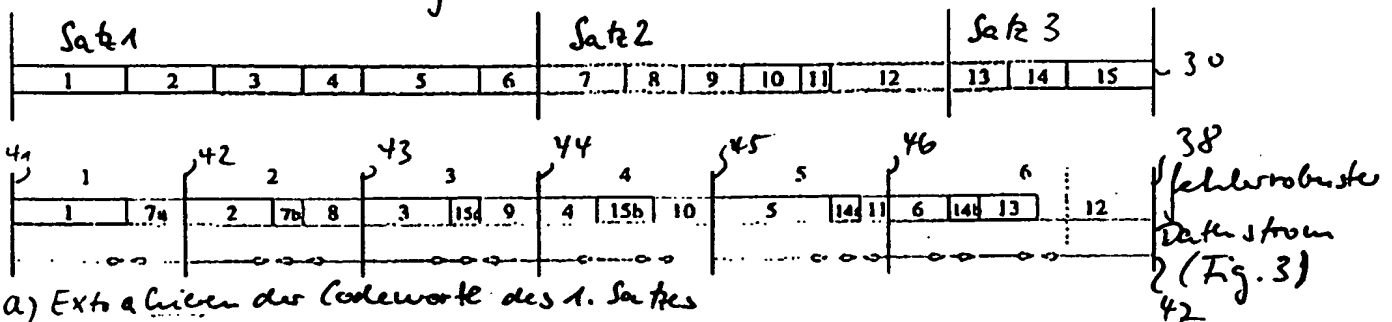
Versuch 6 (13 in 6)



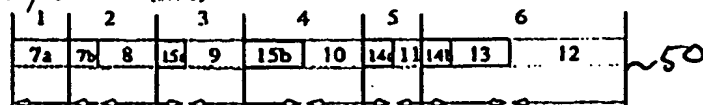
fehlerrobuster Datenstrom

Fig. 3

Codeworte variabler Länge

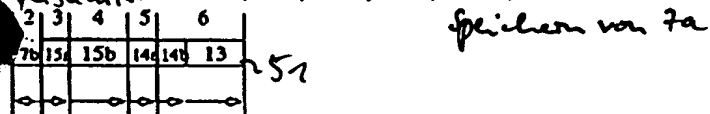


a) Extrahieren der Codeworte des 1. Satzes

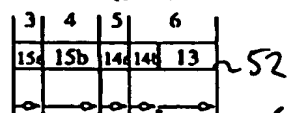


b) Extrahieren der Codeworte des 2. Satzes

Versuch 1 (Suchen von 7 in 1, 8 in 2, 9 in 3, 10 in 4, 11 in 5, 12 in 6)

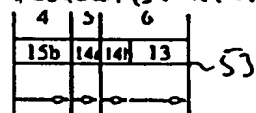


Versuch 2 (Suchen von 7 in 2)

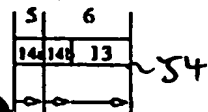


c) Extrahieren der Codeworte des 3. Satzes

Versuch 1 (Suchen von 13 in 1, 14 in 2, 15 in 3)

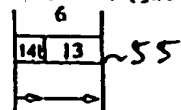


Versuch 2 (Suchen von 13 in 2, 14 in 3, 15 in 4)

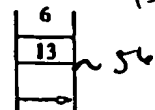


Versuch 3 (Suchen von 13 in 3, 14 in 4)

Versuch 4 (Suchen von 13 in 4, 14 in 5)



Versuch 5 (Suchen von 13 in 5, 14 in 6)

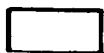


Versuch 6 (Suchen von 13 in 6)

Fig. 4



Prioritätscode worte



Nicht-Prioritätscode worte

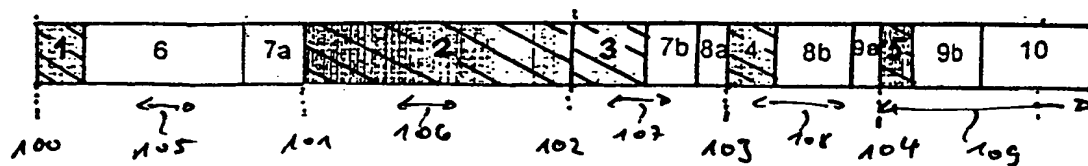


Fig. 5 (Stand der Technik)

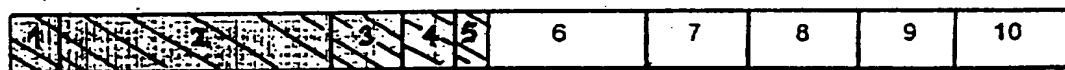


Fig. 6 (Stand der Technik)